RIM: Robust Intersection Management for Connected Autonomous Vehicles

Mohammad Khayatian, Mohammadreza Mehrabian, Aviral Shrivastava
Arizona State University
Why Automated intersections?

- Accidents at intersection
  - Around 30% of fatal crashes have happened in intersection areas, most of which, involved human errors. [FHA]

- Traffic Congestion
  - On the average, each person in the US spends around 42 hours per year stuck in the traffic [FHA]

- As cars become autonomous, so too can intersections
  - Intersection Management using V2I

[FHA] Roadway Safety Data Dashboards, US Department of Transportation - Federal Highway Administration
Intersection Manager (IM)

- An Infrastructure communicating with incoming vehicles.
- Multiple computing systems
- Real-time communication
- Real-time actuation
- Robustness is very important!
Safety buffer

- Localization of autonomous vehicle is challenging!
- IM should consider a safety buffer around each vehicle to account for uncertainties in vehicles position.
Existing Approaches

- **Query-based Intersection Management (QB-IM)**
  - Vehicles send estimated time of arrival and velocity of arrival to IM
  - IM accepts/rejects the request
  - AIM (Autonomous Intersection Management) [1]

- **Velocity Assignment Intersection Management (VA-IM)**
  - Vehicles send their position and velocity
  - A target velocity is assigned to the vehicle
  - Cooperative Vehicle Intersection Control (CVIC) [2]


QB-IM (Query-based Intersection Management)

Intersection Manager

Request Line

Enter  Exit

Not efficient

VA-IM

*(Velocity Assignment Intersection Management)*

Intersection Manager

\[ V_T = 8.5 \text{ m/s} \]

Better Efficiency

Timing Problem in VA-IM

- Round Trip Delay (RTD) is ignored.
  - Inconsistency between vehicle position and what IM thinks

![Diagram of timing problem in VA-IM]

- Worst-case error in position: \( RTD \times V_0 \)
Ignoring RTD Will Cause Crashes

Intersection Manager

\[ V_T = 9.5 \]

Accidents will happen!
Crossroads Technique

- IM sets the execution location to be a fixed value as:

\[
\text{ActuationPosition} = \text{Request Position} + (V_{\text{max}} \times WCRTD) + b
\]

Issues of Crossroads

- **Safety-related**
  - 1) Vehicles are assumed to have zero actuation time
  - 2) Assigned velocity will be maintained until entering the intersection
  - What if a disturbance is applied to the vehicle?
    - Wind, bump, etc.

- **Performance**
  - Vehicles that intend to make a turn will slow down others.
RIM Overview

Phase 1: Sync
Phase 2: Request
Phase 3: TOA & VOA
Phase 4: Track the trajectory

Sync, Request, TOA & VOA

Find the optimal trajectory

Velocity

VOA

Phase 1

Phase 2

Phase 3

Phase 4

TOA

Intersection Manager

Request

Response

Web page: aviral.lab.asu.edu
Our Approach: RIM

Vehicle

1) Clock Sync
3) Send a request to IM
8) Find an optimal trajectory
9) Inform IM
9) Track the trajectory

IM

2) Clock Sync
4) Checks for conflicts
5) Check the feasibility
6) Send TOA and VOA

Algorithm 1: Vehicle Controller
1. If Sync line is crossed then
2. result = synchroniz();
3. if result is not OK then
4. if distance to transmit line is less than d_min then
5. update(Trajectory, SD); /* slow down */
6. end
7. Goto Line 3;
8. end
9. end
10. if Transmit line is crossed then
11. V-Info = [P, V, a, TS, LO, a_max, a_min, ID];
12. send(V-Info);
13. Wait for the response;
14. if response is timed out then
15. if distance to intersection is less than d_min then
16. update(Trajectory, SD); /* slow down */
17. end
18. Goto line 12;
19. end
20. else
21. [TOA, VOA] = getPacket(response);
22. [a_0, b_0] = calculateTrajectory(TOA, VOA);
23. update(Trajectory, [a_0, b_0]); /* set the Ref Trajectory */
24. end

Algorithm 2: IM’s Scheduling algorithm
1. Input: Request;
2. Outputs: [TOA, VOA];
3. while Request buffer is not empty do
4. V-Info = read(buffer(first));
5. [TOA, VOA] = Schedule(V-Info, I-Info);
6. Result = F-Check(TOA, VOA, V-Info, I-Info);
7. If Result is OK then
8. Send(TOA, VOA, Vehicle Info);
9. update(I-Info)
10. else
11. Increase(TOA);
12. Goto Line 6;
13. end
14. end
Find the Optimal Trajectory

- We define a functional based on acceleration:
  \[ J = \int_{t_0}^{t_f} a^2 \, dt \]

- Solve using Fundamental Lemma of the Calculus Variation:
  \[ a(t) = A_0 t + B_0 \]

- Linear acceleration is optimal!
Find the Optimal Trajectory (cont.)

- Taking integral from acceleration we have:
  \[ v(t) = \frac{1}{2} A_0 t^2 + B_0 t + v_0 \]
- Taking integral from velocity we have:
  \[ x(t) = \frac{1}{6} A_0 t^3 + \frac{1}{2} B_0 t^2 + v_0 t + x_0 \]
- Substituting the initial and final conditions \((x(t_0) = x_0, x(t_f) = x_f, v(0) = v_0, v(t_f) = v_f)\), we have:
  \[ v(t_f) = \frac{1}{2} A_0 t_f^2 + B_0 t_f + v_0 \]
  \[ x(t_f) = \frac{1}{6} A_0 t_f^3 + \frac{1}{2} B_0 t_f^2 + v_0 t_f + x_0 \]
  \[ A_0 = \frac{6(2x_0 - 2x_f + t_f v_0 + t_f v_f)}{t_f^3} \]
  \[ B_0 = \frac{-2(3x_0 - 3x_f + 2t_f v_0 + t_f v_f)}{t_f^2} \]
Dealing with the Round-trip Delay

- As soon as TOA and VOA are received, the optimal trajectory is determined.

- All determined trajectories will meet the TOA and VOA requirement.

- Vehicles will control the position instead of velocity.
Feasibility of VOA and TOA

- Is the assigned TOA and VOA feasible?

- Check for min and max acceleration rates
  - Vehicles have limited acc/dec power

- Check for min and max velocities
  - Road speed limit

Feasibility Check for A Set of TOA and VOA
Feasibility of VOA and TOA

Feasibility Analysis
Checks for Inter-trajectory Conflicts

Position (m)

0  2  4  6  8  10  12

Velocity (m/s)

0  2  4  6  8  10  12

Request line

Feasible TOA
Infeasible TOA
Minimum Distance

Front Car
Rear car - infeasible trajectory
Rear car - feasible trajectory

Front Car
Min speed
Max speed
Rear car - infeasible trajectory
Rear car - feasible trajectory

Web page: aviral.lab.asu.edu
Our Testbed

- 4-way intersection
- 1/10 scale RC cars on Traxxas chassis
- Vehicle size = 30 cm x 57 cm
  max speed = 5 m/s (11.1 mph)
- Lane width = 60 cm
- Transmit line distance = 3 m
- NTP protocol for clock synchronization (10 ms accuracy)
- PID controller for position trajectory tracking
- 3 types of communication packets
  - Sync (7 Bytes), Request (30 Bytes), Receive (30 Bytes)
IM & Car Schematic

IM
- IM uC
- Transceiver
- SPI

Wireless

Autonomous Car
- Main uC
- Transceiver
- SPI

Motor
- PWM

1st uC
- I²C
- USART
- LIDAR

IM uC
- SPI
- USART
- Square Wave

2nd uC
- I²C
- Motor
- Encoder

IM & Car
- IM uC
- Transceiver
- SPI

Web page: aviral.lab.asu.edu
Video of our Intersection
10% model mismatch is added.

An external disturbance of 5% is applied.

Monitor position and velocity of a CAV.
RIM is Robust

- 10% model mismatch is added.
- An external disturbance of 5% is applied.
- Monitor position and velocity of a CAV.

![Graph showing position and velocity over time with an almost zero error and disturbance is applied at specific times.](image)
Robustness of RIM vs Crossroads

- We repeated our experiment 50 times, for different VOAs and TOAs.
Crossroads needs a Safety Buffer

- Since Crossroads does not consider model mismatches and external disturbances, it’s not safe!

- However, Crossroads can still work if we consider a safety buffer of 3.3X of size of the car.
Evaluation using our Simulator

- Developed in MATLAB® [1]
- Scale to multi-lane intersection
- Variable traffic flow rates
- Model vehicle dynamics
- Model network delay

Simulator Video

Available Online: https://github.com/mkhayatian/Traffic-Intersection-Simulator-for-Autonomous-Vehicles

Web page: aviral.lab.asu.edu
Throughput Improvement

COMPARING WITH CROSSROADS
INCREASE IN THROUGHPUT FOR DIFFERENT FLOW RATES

IMPROVEMENT IN THROUGHPUT
(percentage)

FLOW RATE (VEHICLE PER LANE PER SECOND)

- 3 Lane intersection
- Single lane intersection
Crossroads & VA-IM
- A **constant** Velocity is assigned.
- To avoid rollover, turn speed is low.
- Vehicles that intend to make a turn will slow down behind vehicles.

RIM
- Vehicles can adjust their speed.

**Speedup in Throughput for different turn velocity limits**

<table>
<thead>
<tr>
<th>Turn Velocity Limit (M/S)</th>
<th>RIM</th>
<th>Crossroads</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>8</td>
<td>0.4</td>
</tr>
<tr>
<td>0.6</td>
<td>6.5</td>
<td>0.6</td>
</tr>
<tr>
<td>0.8</td>
<td>5.5</td>
<td>0.8</td>
</tr>
<tr>
<td>1</td>
<td>4.5</td>
<td>1.0</td>
</tr>
<tr>
<td>1.2</td>
<td>3.5</td>
<td>1.2</td>
</tr>
<tr>
<td>1.4</td>
<td>2.5</td>
<td>1.4</td>
</tr>
<tr>
<td>AVG</td>
<td>1.5</td>
<td>0.2</td>
</tr>
</tbody>
</table>

**Efficiently Making a Turn**

Web page: aviral.lab.asu.edu
Position of Our Work

- Query-based (AIM)
- Velocity Assignment
- Crossroads
- RIM
Conclusion

- We explored safety concerns of previous intersection management techniques.
  - Network delay
  - Vehicle dynamics

- We presented a **robust** intersection management interface for connected autonomous vehicles.
  - Model mismatch
  - External disturbances
  - Efficient
Question?